

In the United States Patent and Trademark Office
Board of Patent Appeals and Interferences

Appeal Brief

In re the Application of:

Laurence E. England, Howard J. Glaser, and Robert D. Moyer

Serial No. 09/591,035
Filed: June 9, 2000
Attorney Docket No. STL920000063US1

METHOD OF, SYSTEM FOR, AND COMPUTER PROGRAM PRODUCT FOR
PROVIDING AN E-BUSINESS CONNECTOR BUILDER

Submitted by:

Konrad, Raynes & Victor LLP
315 So. Beverly Dr., Ste. 210
Beverly Hills CA 90212
(310) 556-7983
(310) 556-7984 (fax)

TABLE OF CONTENTS

I. Real Party in Interest.....	1
II. Related Appeals, Interferences, and Judicial Proceedings.....	1
III. Status of the Claims	1
IV. Status of Amendments	1
V. Summary of the Claimed Subject Matter.....	1
VI. Grounds of Rejection to Be Reviewed on Appeal.....	6
VII. Argument	7
A. Rejection Under 35 U.S.C. §103(a) Over Helgeson et al. (U.S. 2002/0073236, hereinafter “Helgeson”) and in view of O’Brien et al. (U.S. 6,351,776, hereinafter “O’Brien”).....	7
1. Claims 1-24	7
2. Conclusion	17
VIII. Claims Appendix	18
IX. Evidence Appendix	26
X. Related Proceedings Appendix	27

I. Real Party in Interest

The entire right, title and interest in this patent application is assigned to real party in interest International Business Machines Corporation.

II. Related Appeals, Interferences, and Judicial Proceedings

Appellant, Appellant's legal representative, and Assignee are not aware of any other prior or pending appeals, interferences, and judicial proceedings which may be related to, directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. Status of the Claims

Claims 1-24 are pending and have been rejected in view of prior art. The final rejection of the claims is being appealed for all pending claims 1-24.

IV. Status of Amendments

A Final Office Action was mailed on May 18, 2006. In response to the Final Office Action, a Notice of Appeal was filed on August 17, 2006, without making any amendments.

V. Summary of the Claimed Subject Matter

Independent claim 1 describes a computer-implemented method of adapting a transaction-based mainframe application to process transactions over a network (e.g., Specification, page 10, line 23 – page 11, line 1; FIG. 1). The transaction-based

mainframe application comprises source code describing a transaction and information related to the transaction, hereinafter related information (e.g., Specification, page 11, lines 13-17; page 14, lines 2-4). The source code of the transaction-based mainframe application is scanned to identify the transaction and the related information, wherein the transaction-based mainframe application is unable to process transactions over a World Wide Web (Web) (e.g., Specification, page 10, lines 23-26; page 14, lines 2-4). The related information identified in the scan of the source code, hereinafter identified information, is stored in a database (e.g., Specification, page 11, lines 13-19; page 14, lines 4-5). Parameter definitions describing a communication of information by the transaction, hereinafter extracted information, are extracted from the database (e.g., Specification, page 14, lines 5-6). A parameter usage type for each parameter is identified, said parameter usage type selectable from the parameter usage type set comprising input, output, input/output, and unreferenced (e.g., Specification, page 14, lines 6-8). The transaction and a subset of the related information and extracted information is displayed (e.g., Specification, page 14, lines 8-9). A user is allowed to select the transaction (e.g., Specification, page 14, lines 9-10). The identified information and extracted information are used to package the user-selected transaction in a form compatible with a connector building tool by generating a communication area file that may be parsed by the connector building tool to build a connector and a documentation file that provides documentation about the communication area file, wherein the connector enables the transaction-based mainframe application to process transactions over the web by enabling the transactions to be passed from a web application server to

the transaction-based mainframe application (e.g., Specification, page 12, line 27 – page 13, line 9; page 14, lines 10-14).

Independent claim 9 describes an article of manufacture for use in a computer system for adapting a transaction-based mainframe application to process transactions over a network (e.g., Specification, page 10, line 23 – page 11, line 1; page 14, lines 18-26; FIG. 1). The transaction-based mainframe application comprises source code describing a transaction and information related to the transaction, hereinafter related information (e.g., Specification, page 11, lines 13-17; page 14, lines 2-4). The article of manufacture comprises a computer-readable storage medium having a computer program embodied in said medium which causes the computer system to execute steps (e.g., Specification, page 14, lines 18-26). The source code of the transaction-based mainframe application is scanned to identify the transaction and the related information, wherein the transaction-based mainframe application is unable to process transactions over a World Wide Web (Web) (e.g., Specification, page 10, lines 23-26; page 14, lines 2-4). The related information identified in the scan of the source code, hereinafter identified information, is stored in a database (e.g., Specification, page 11, lines 13-19; page 14, lines 4-5). Parameter definitions describing a communication of information by the transaction, hereinafter extracted information, are extracted from the database (e.g., Specification, page 14, lines 5-6). A parameter usage type is identified for each parameter, said parameter usage type selectable from the parameter usage type set comprising input, output, input/output, and unreferenced (e.g., Specification, page 14, lines 6-8).

The transaction and a subset of the related information and extracted information are displayed (e.g., Specification, page 14, lines 8-9). A user is allowed to select the transaction(e.g., Specification, page 14, lines 9-10). The identified information and extracted information are used to package the user-selected transaction in a form compatible with a connector building tool by generating a communication area file that may be parsed by the connector building tool to build a connector and a documentation file that provides documentation about the communication area file, wherein the connector enables the transaction-based mainframe application to process transactions over the web by enabling the transactions to be passed from a web application server to the transaction-based mainframe application (e.g., Specification, page 12, line 27 – page 13, line 9; page 14, lines 10-14).

Independent claim 17 describes a computer system for adapting a transaction-based mainframe application to process transactions over a network ((e.g., Specification, page 9, lines 12-13; page 10, line 23 – page 11, line 1; page 14, lines 16-18; FIG. 1). The transaction-based mainframe application comprises source code describing a transaction and information related to the transaction, hereinafter related information (e.g., Specification, page 11, lines 13-17; page 14, lines 2-4). A scanner is able to scan the source code of the transaction-based mainframe application to identify the transaction and the related information, wherein the transaction-based mainframe application is unable to process transactions over a World Wide Web (Web) (e.g., Specification, page 10, lines 23-26; page 14, lines 2-4). Storage is able to store in a database the related information identified in the scan of the source code, hereinafter identified information (e.g., Specification, page 11, lines 13-19; page 14, lines 4-5). A query is able to extract

from the database parameter definitions describing a communication of information by the transaction, hereinafter extracted information (e.g., Specification, page 14, lines 5-6). An identifying computer program is able to identify a parameter usage type for each parameter, said parameter usage type selectable from the parameter usage type set comprising input, output, input/output, and unreferenced (e.g., Specification, page 14, lines 6-8). A display is able to display the transaction and a subset of the related information and extracted information (e.g., Specification, page 14, lines 8-9). An interface is able to allow a user to select the transaction (e.g., Specification, page 14, lines 9-10). A packaging computer program is able to use the identified information and extracted information to package the user-selected transaction in a form compatible with a connector building tool by generating a communication area file that may be parsed by the connector building tool to build a connector and a documentation file that provides documentation about the communication area file, wherein the connector enables the transaction-based mainframe application to process transactions over the web by enabling the transactions to be passed from a web application server to the transaction-based mainframe application (e.g., Specification, page 12, line 27 – page 13, line 9; page 14, lines 10-14).

Claims 4, 12, and 20 describe that the documentation file comprises field description information and connection information (e.g., Specification, page 13, lines 3-5).

Claims 5, 13, and 21 describe using the identified information and extracted information to build a connector (e.g., Specification, page 13, lines 1-3; page 14, lines 13-14).

Claims 6, 14, and 22 describe using the identified information and extracted information to build an enterprise Java bean connector (e.g., Specification, page 5, line 29 – page 6, line 3).

Claims 7, 15, and 23 describe that the database can be queried to find program parts comprising the transaction-based mainframe application and identify relationships between the program parts (e.g., Specification, page 11, lines 13-19).

Claims 8, 16, and 24 describe that the related information is a member of the set comprising relationships, call hierarchies, transactions, communication areas, parameters, the flow of data elements, and resources employed (e.g., Specification, page 11, lines 13-19).

VI. Grounds of Rejection to Be Reviewed on Appeal

A concise statement listing each ground of rejection presented for review is as follows:

Whether claims 1-24 are unpatentable under 35 U.S.C. 103(a) over Helgeson et al. (U.S. 2002/0073236, hereinafter “Helgeson”) and in view of O’Brien et al. (U.S. 6,351,776, hereinafter “O’Brien”).

VII. Argument

- A. Rejection Under 35 U.S.C. §103(a) Over Helgeson et al. (U.S. 2002/0073236, hereinafter “Helgeson”) and in view of O’Brien et al. (U.S. 6,351,776, hereinafter “O’Brien”)

1. Claims 1-24

Independent claim 1 describes a computer-implemented method of adapting a transaction-based mainframe application to process transactions over a network, wherein the transaction-based mainframe application is unable to process transactions over a World Wide Web (Web).

That is, Applicants' claimed invention is directed towards analyzing a transaction-based mainframe application in order to build a connector to adapt the transaction-based mainframe application to process transactions over a network such as the Web. That is, the claimed invention is directed to retrofitting/adapting existing mainframe applications to support customer to business transactions (C2B) as well as business to business transactions (B2B) on the web (e.g., Specification, page 3, lines 11-13). To enable current mainframe applications and data stores to support such transactions, the claimed invention provides a technique to pass transactions from web application servers to back end applications residing on the mainframe (e.g., Specification, page 3, lines 15-17).

The Helgeson patent application translates data from a system specific local format to a generic interchange format object, and vice versa, with predefined stylesheets using generic components and system specific service components that utilize a native application programming interface of the specific local system. (Abstract). Moreover, the Helgeson patent application describes a system that "is predominantly web-enabled,

which extends its use to all industry professionals connected to the Internet". (Paragraph 42). The Helgeson patent application describes a Platform that "provides a unified set of interfaces, an application Framework, Web-application development, external connectivity development, and information distribution development." (Paragraph 42). The Platform model 501 defines applicants' application platform and includes a Web Development Kit (WDK) server that generates web content (Paragraphs 63-65). The Helgeson patent application describes a three-tier model in which a tier 1 web user is connected electronically to a tier 2 web server which is connected to a tier 3 applications server. (Paragraph 214; FIG. 3). The tier 3 applications server is expanded in FIG. 4 to show servers, including an Interface server (also designated as the WDK), which communicates through a web server via the internet to web clients via the HTML protocol and which contains mechanisms to manipulate various kinds of display style sheets, to generate and execute web links, . . . (Paragraph 215; FIG. 4). Because the Helgeson patent application is predominantly web-enabled, there is no need to enable a transaction-based mainframe application that is not originally designed to process transactions over the web to do so.

Moreover, in the Helgeson patent application describes that its system is, and will be, able to interface with any other industry standard software programs (Paragraph 41). Thus, the Helgeson patent application again has no need to perform the claimed processing that creates a connector to enable the transaction-based mainframe application to process transactions over the web by enabling the transactions to be passed from a web application server to the transaction-based mainframe application.

In particular, claim 1 describes scanning the source code of the transaction-based mainframe application to identify the transaction and the related information, wherein the transaction-based mainframe application is unable to process transactions over a World Wide Web (Web). The Examiner cites page 2, paragraph 16 of the Helgeson patent application as teaching this. Applicants traverse. Page 2, paragraph 0016 of the Helgeson patent recites:

The system may also include a monitor component for monitoring changes of a data object at a system, with the monitoring component having both a system independent service subcomponent and a system specific service component utilizing a native API of the monitored system to monitor changes of the data object. The system may also include a mapper component for identifying a local object identifier and a document type.

That is, paragraph 16 of the Helgeson patent application describes monitoring changes of a data object. For example, the monitor 945 monitors changes to local objects and reports changes to interested parties, and clients can register to receive notification of the change only or have the changed object sent with the notification (page 50, paragraph 847). Applicants submit that monitoring changes of a data object does not teach or suggest scanning the source code of the transaction-based mainframe application to identify the transaction and the related information.

As discussed above, Applicants submit that the Helgeson patent teaches a system that is web-enabled, which teaches away from having a transaction-based mainframe application is unable to process transactions over a World Wide Web (Web). However, the Examiner cites page 3, paragraph 39 of the Helgeson patent as teaching that the

transaction-based mainframe application is unable to process transactions over a World Wide Web (Web). Applicants traverse. Paragraph 39 of the Helgeson patent describes a Business Management System Platform Architecture that is designed to maintain and use a set of unique servers and common objects to generate the set of tasks required to be performed to complete a designated business transaction in a concrete, and useful way. Paragraph 39 goes on to describe that the platform permits application developers to work on the business aspects of the application without having to focus on transaction management, security, persistence of data or life cycle management of the objects itself. There is no indication in paragraph 39 that a business application does not process transactions over a network. Thus, Applicants submit that paragraph 39 does not teach or suggest that a transaction-based mainframe application is unable to process transactions over a World Wide Web (Web). Instead, the Helgeson system is predominantly web-enabled (page 3, paragraph 42), which teaches away from a transaction-based mainframe application that is unable to process transactions over a World Wide Web (Web).

In addition, claim 1 describes storing in a database the related information identified in the scan of the source code, hereinafter identified information. The Examiner cites page 19, paragraphs 381-382 of the Helgeson patent application as teaching this. Applicants traverse. Paragraphs 381-382 describe persistence of data that allows the bean's definition to be independent of the type of data store used for persistence and describe that a component developer will be responsible for declaring part or all of the attributes of an entity bean as persistent in its deployment descriptor and then mapping them to fields in a database at deployment time. Merely storing data persistently does not teach or suggest storing in a database the related information

identified in the scan of the source code. Also, as discussed above, the Helgeson patent application fails to teach scanning application source code to identify a transaction and related information related to the transaction, and so the Helgeson patent application also fails to teach storing the related information identified in the scan of the source code. There is no teaching or suggestion in the Helgeson patent application that the stored data is related information that was identified by scanning the source code of a transaction-based mainframe application that is unable to process transactions over a World Wide Web (Web).

Moreover, claim 1 describes identifying a parameter usage type for each parameter, said parameter usage type selectable from the parameter usage type set comprising input, output, input/output, and unreferenced. The Examiner cites page 12, paragraphs 277-278, page 48, paragraph 836, and page 19, paragraph 387 of the Helgeson patent as teaching this. Applicants traverse. Paragraphs 277-278 describe invocation of database stored procedures that store arguments in specific fields in a data base and vice versa. Paragraph 387 recites:

For transactions, an application developer has two options: 1) to explicitly demarcate the boundaries of a transaction, or 2) to use declarative transactional management available with EJBs. Use of declarative transactional management is cleaner and is strongly recommended. In this case, the level of granularity for managing transactions corresponds to methods in a bean. Instead of interleaving transaction boundaries within business logic, transactional attributes are separately declared in the bean's deployment descriptor (for a specific method, or as the bean's default) as one of the following six options:

TX_NOT_SUPPORTED, TX_SUPPORTS, TX_REQUIRED,
TX_REQUIRES_NEW, TX_MANDATORY, TX_BEAN_MANAGED. Details
of these can be found in books on EJB.

That is, paragraph 387 describes that transactional attributes that are separately declared in the bean's deployment descriptor as one of the following six options:

TX_NOT_SUPPORTED, TX_SUPPORTS, TX_REQUIRED, TX_REQUIRES_NEW,
TX_MANDATORY, TX_BEAN_MANAGED. Applicants submit that transactional attributes do not teach or suggest describes identifying a parameter usage type for each parameter, said parameter usage type selectable from the parameter usage type set comprising input, output, input/output, and unreferenced.

Claim 1 also describes displaying the transaction and a subset of the related information and extracted information. The Examiner cites page 19, paragraph 387 of the Helgeson patent as teaching this. Applicants traverse. Paragraph 387 describes that for transactions, an application developer has two options: 1) to explicitly demarcate the boundaries of a transaction, or 2) to use declarative transactional management available with EJBs. Such options do not teach or suggest displaying the transaction and a subset of the related information and extracted information.

Claim 1 describes allowing a user to select the transaction. The Examiner submits that the Helgeson patent application does not disclose this.

Additionally, claim 1 describes using the identified information and extracted information to package the user-selected transaction in a form compatible with a connector building tool by generating a communication area file that may be parsed by the connector building tool to build a connector and a documentation file that provides

documentation about the communication area file. The Examiner cites the O'Brien patent at col. 15, lines 21-27 as teach this. Applicants traverse. The cited portion of the O'Brien patent recites:

 If at step 714 the user must be sent back to the same database, query is made at step 740 to determine if that database is still up. If it is, the request is passed to the pool specification 720 where it is subsequently passed to the database object 236, on to the connection pool 730, and the appropriate database, either the transaction database 150 or the query database 152.

The O'Brien patent fails to teach or suggest using identified information and extracted information to package a user-selected transaction in a form compatible with a connector building tool, wherein the identified information and extracted information enable building a connector to enable the transaction-based mainframe application to process transactions over the network. A "connection pool" is not a connector building tool, and the O'Brien patent teaches at col. 8, lines 16-18 that the "EJB cluster (EJBC) caches memory of common resources such as the pooling of data connections and the like, as well as data objects." Thus, a connection pool is the pooling of data connections, rather than the claimed connector building tool. The claimed connector building tool may be, for example, the IBM Enterprise Access Beans (EAB) or the Microsoft COMTI Builder (e.g., Specification, page 11, lines 7-9 and lines 24-26). Also, passing a request to a pool specification and subsequently to a database does not teach or suggest packaging a user-selected transaction in a form compatible with a connector building tool.

In addition, the O'Brien patent provides means by which users can establish, use, and maintain files on the Internet in a manner remote from their local computers

(Abstract). The O'Brien patent is not attempting to process the source code of a transaction-based mainframe application that does not process transactions over the network to output data in a form compatible with a connector building tool to enable the transaction-based mainframe application to process transactions over the network.

Furthermore, claim 1 describes that the connector enables the transaction-based mainframe application to process transactions over the web by enabling the transactions to be passed from a web application server to the transaction-based mainframe application. The Examiner cites page 21, paragraph 420 and page 22 paragraph 424 of the Helgeson patent as teaching this. Applicants traverse. Paragraph 420 describes maintaining transactional integrity using the transactional attribute of TX_REQUIRED for a method cancelClass() in the bean's deployment descriptor. Paragraph 424 describes that EJBs are packaged as EJB jar files that are comprised of the class files for the bean class, the home interface, the remote interface, the primary key class (if applicable), in addition to the deployment descriptor and a manifest. Applicants submit that the cited portions of the Helgeson patent do not address the claimed connector, which is built using a communication area file that may be parsed by the connector building tool.

Moreover, the Examiner submits that the O'Brien and Helgeson combination makes it efficient for files to be available worldwide through the Internet. Applicants submit that the claimed invention is not directed to making files available worldwide. Instead, the claimed invention is directed to processing the source code of a transaction-based mainframe application that does not process transactions over the network to output data in a form compatible with a connector building tool, wherein identified

information and extracted information enable building a connector to enable the transaction-based mainframe application to process transactions over the network.

Applicants submit that a connection pool does not teach or suggest using the identified information and extracted information to package the user-selected transaction in a form compatible with a connector building tool by generating a communication area file that may be parsed by the connector building tool to build a connector and a documentation file that provides documentation about the communication area file.

Applicants submit that the combination of the Helgeson patent application and the O'Brien patent fails to teach or suggest processing the source code of a transaction-based mainframe application that is unable to process transactions over a World Wide Web (Web) to use identified information and extracted information to package the user-selected transaction in a form compatible with a connector building tool by generating a communication area file that may be parsed by the connector building tool to build a connector and a documentation file that provides documentation about the communication area file, wherein the connector enables the transaction-based mainframe application to process transactions over the web by enabling the transactions to be passed from a web application server to the transaction-based mainframe application.

Thus, the subject matter of independent claim 1 is not taught or suggested by the Helgeson patent application or the O'Brien patent, either alone or in combination.

The subject matter of claims 9 and 17 are not taught or suggested by the Helgeson patent application or the O'Brien patent, either alone or in combination, at least for the reasons discussed with reference to claim 1.

Dependent claims 2-8, 10-16, and 18-24 incorporate the language of independent claims 1, 9, and 17 and add additional novel elements. Therefore, dependent claims 2-8, 10-16, and 18-24 are not taught or suggested by the Helgeson patent application or the O'Brien patent, either alone or in combination, for at least the same reasons as were discussed with respect to claims 1, 9, and 17.

In addition, claims 5, 13, and 21 describe using the identified information and extracted information to build a connector. Claims 6, 14, and 22 describe using the identified information and extracted information to build an enterprise Java bean connector. The Examiner cites Col. 15, lines 21-41 of the O'Brien patent as teaching these claims. Applicants traverse. The cited portion of the O'Brien patent describes a connection pool, but, there is no teaching or suggestion of using the identified information and extracted information to build a connector or to build an enterprise Java bean connector.

Accordingly, it is respectfully submitted that the rejection of claims 1-24 as obvious over the Helgeson and O'Brien combination should be reversed.

2. Conclusion

Each of the rejections set forth in the Final Office Action is improper and should be reversed.

Respectfully submitted,

/Janaki K. Davda/

Dated: October 17, 2006

Janaki K. Davda
Reg. No. 40,684
Konrad Raynes & Victor LLP
315 South Beverly Drive, Ste. 210
Beverly Hills, California 90212
Tel: 310-553-7977
Fax: 310-556-7984

VIII. Claims Appendix

1. (Previously Presented) A computer-implemented method of adapting a transaction-based mainframe application to process transactions over a network, said transaction-based mainframe application comprising source code describing a transaction and information related to the transaction, hereinafter related information, said method comprising:

scanning the source code of the transaction-based mainframe application to identify the transaction and the related information, wherein the transaction-based mainframe application is unable to process transactions over a World Wide Web (Web);

storing in a database the related information identified in the scan of the source code, hereinafter identified information;

extracting from the database parameter definitions describing a communication of information by the transaction, hereinafter extracted information;

identifying a parameter usage type for each parameter, said parameter usage type selectable from the parameter usage type set comprising input, output, input/output, and unreferenced;

displaying the transaction and a subset of the related information and extracted information;

allowing a user to select the transaction; and

using the identified information and extracted information to package the user-selected transaction in a form compatible with a connector building tool by generating a communication area file that may be parsed by the connector building tool to build a connector and a documentation file that provides documentation about the

communication area file, wherein the connector enables the transaction-based mainframe application to process transactions over the web by enabling the transactions to be passed from a web application server to the transaction-based mainframe application.

2. (Previously Presented) The method of claim 1 wherein the communication area file contains information which can be parsed by a connector building tool.

3. (Previously Presented) The method of claim 2 wherein the documentation file describes the communication area file.

4. (Original)The method of claim 3 wherein the documentation file comprises field description information and connection information.

5. (Previously Presented) The method of claim 1 further comprising using the identified information and extracted information to build a connector.

6. (Previously Presented) The method of claim 5 further comprising using the identified information and extracted information to build an enterprise Java bean connector.

7. (Previously Presented) The method of claim 1 wherein the database can be queried to find program parts comprising the transaction-based mainframe application and identify relationships between the program parts.

8. (Original) The method of claim 1 wherein the related information is a member of the set comprising relationships, call hierarchies, transactions, communication areas, parameters, the flow of data elements, and resources employed.

9. (Previously Presented) An article of manufacture for use in a computer system for adapting a transaction-based mainframe application to process transactions over a network, said transaction-based mainframe application comprising source code describing a transaction and information related to the transaction, hereinafter related information, said article of manufacture comprising a computer-readable storage medium having a computer program embodied in said medium which causes the computer system to execute the method steps comprising:

scanning the source code of the transaction-based mainframe application to identify the transaction and the related information, wherein the transaction-based mainframe application is unable to process transactions over a World Wide Web (Web);

storing in a database the related information identified in the scan of the source code, hereinafter identified information;

extracting from the database parameter definitions describing a communication of information by the transaction, hereinafter extracted information;

identifying a parameter usage type for each parameter, said parameter usage type selectable from the parameter usage type set comprising input, output, input/output, and unreferenced;

displaying the transaction and a subset of the related information and extracted information;

allowing a user to select the transaction; and

using the identified information and extracted information to package the user-selected transaction in a form compatible with a connector building tool by generating a communication area file that may be parsed by the connector building tool to build a connector and a documentation file that provides documentation about the communication area file, wherein the connector enables the transaction-based mainframe application to process transactions over the web by enabling the transactions to be passed from a web application server to the transaction-based mainframe application.

10. (Previously Presented) The article of manufacture of claim 9 wherein the communication area file contains information which can be parsed by a connector building tool.

11. (Previously Presented) The article of manufacture of claim 10 wherein the documentation file describes the communication area file.

12. (Original) The article of manufacture of claim 11 wherein the documentation file comprises field description information and connection information.

13. (Previously Presented) The article of manufacture of claim 9 wherein the method steps further comprise using the identified information and extracted information to build a connector.

14. (Previously Presented) The article of manufacture of claim 13 wherein the method steps further comprise using the identified information and extracted information to build an enterprise Java bean connector.

15. (Previously Presented) The article of manufacture of claim 9 wherein the database can be queried to find program parts comprising the transaction-based mainframe application and identify relationships between the program parts.

16. (Original) The article of manufacture of claim 9 wherein the related information is a member of the set comprising relationships, call hierarchies, transactions, communication areas, parameters, the flow of data elements, and resources employed.

17. (Previously Presented) A computer system for adapting a transaction-based mainframe application to process transactions over a network, said transaction-based mainframe application comprising source code describing a transaction and information related to the transaction, hereinafter related information, said computer system comprising :

a scanner for scanning the source code of the transaction-based mainframe application to identify the transaction and the related information, wherein the

transaction-based mainframe application is unable to process transactions over a World Wide Web (Web);

storage for storing in a database the related information identified in the scan of the source code, hereinafter identified information;

a query for extracting from the database parameter definitions describing a communication of information by the transaction, hereinafter extracted information;

an identifying computer program for identifying a parameter usage type for each parameter, said parameter usage type selectable from the parameter usage type set comprising input, output, input/output, and unreferenced;

a display for displaying the transaction and a subset of the related information and extracted information;

an interface allowing a user to select the transaction; and

a packaging computer program which uses the identified information and extracted information to package the user-selected transaction in a form compatible with a connector building tool by generating a communication area file that may be parsed by the connector building tool to build a connector and a documentation file that provides documentation about the communication area file, wherein the connector enables the transaction-based mainframe application to process transactions over the web by enabling the transactions to be passed from a web application server to the transaction-based mainframe application.

18. (Previously Presented) The computer system of claim 17 wherein the communication area file contains information which can be parsed by a connector building tool.

19. (Previously Presented) The computer system of claim 18 wherein the documentation file describes the communication area file.

20. (Original) The computer system of claim 19 wherein the documentation file comprises field description information and connection information.

21. (Previously Presented) The computer system of claim 17 further comprising a connector builder which uses the identified information and extracted information to build a connector.

22. (Previously Presented) The computer system of claim 21 wherein the connector builder uses the identified information and extracted information to build an enterprise Java bean connector.

23. (Previously Presented) The computer system of claim 17 wherein the database can be queried to find program parts comprising the transaction-based mainframe application and identify relationships between the program parts.

24. (Original) The computer system of claim 17 wherein the related information is a member of the set comprising relationships, call hierarchies, transactions, communication areas, parameters, the flow of data elements, and resources employed.

IX. Evidence Appendix

None

X. Related Proceedings Appendix

None